دليل ربط جهاز مراقبة الجنين BT-350 بالكمبيوتر وعرض التخطيط

المؤلف: Manus AI

1. مقدمة

يهدف هذا الدليل إلى مساعدة المستخدمين في ربط جهاز مراقبة الجنين Bistos BT-350 بجهاز الكمبيوتر وعرض بيانات تخطيط قلب الجنين (FHR) وانقباضات الرحم (UC) بشكل صحيح. سيتناول الدليل المشاكل الشائعة المتعلقة بمقياس الرسم وسرعة الطباعة، ويقدم حلولاً ممكنة، بما في ذلك فهم بروتوكول الاتصال (التشفير) للبيانات المرسلة عبر منفذ RS-232C.

2. فهم المشكلة: التخطيط الطويل وسرعة الطباعة

واجه المستخدم مشكلة رئيسية تتمثل في أن التخطيط الذي يظهر على الكمبيوتر يكون أطول بكثير من التخطيط المطبوع من جهاز BT-350، وأن البرنامج المستخدم لا يقوم بطباعة 1 سم بشكل صحيح، على الرغم من ضبط سرعة الطباعة على '1'. هذه المشكلة تشير إلى عدم تطابق في كيفية تفسير وعرض البيانات بين الجهاز والبرنامج على الكمبيوتر.

2.1. الأسباب المحتملة:

- **عدم تطابق إعدادات مقياس الرسم (Scaling):** أجهزة مراقبة الجنين تطبع التخطيط بمقاييس زمنية قياسية (مثل 1 سم/دقيقة أو 3 سم/دقيقة). إذا كان البرنامج يعرض البيانات بمقياس مختلف، فسيظهر التخطيط أطول أو أقصر من المتوقع. قد يكون البرنامج يستخدم مقياسًا افتراضيًا غير متوافق، أو أن إعدادات المقياس لم يتم ضبطها بشكل صحيح.
- **تفسير خاطئ للبيانات الزمنية:** قد يرسل الجهاز البيانات بتردد معين، وإذا كان البرنامج يفسر هذا التردد بشكل خاطئ، فقد يؤدي ذلك إلى تمدد أو انكماش التخطيط على الشاشة.
- **مشاكل في إعدادات الطباعة:** إذا كانت سرعة الطباعة في البرنامج مضبوطة على '1' (والتي غالبًا ما تعني 1 سم/دقيقة)، ولكن التخطيط المطبوع لا يتوافق مع هذا المقياس، فهذا يشير إلى عدم وجود معايرة صحيحة للطباعة، أو أن البرنامج يعتمد على إعدادات طابعة افتراضية غير مناسبة.

- **برنامج غير متوافق أو غير مخصص:** قد يكون البرنامج المستخدم ليس هو البرنامج المخصص من الشركة المصنعة (BCM-350)، أو أنه برنامج عام لا يدعم بشكل كامل بروتوكول الاتصال الخاص بـ BT-350.
- **مشاكل في كابل الاتصال:** على الرغم من أن البيانات تصل، إلا أن كابل RS-232C غير الصحيح (مثل استخدام كابل مستقيم بدلاً من كابل متصالب - null modem cable) يمكن أن يسبب أخطاء في البيانات أو عدم استقرار في الاتصال، مما يؤثر على العرض الصحيح. [1]

3. الحلول المقترحة

3.1. استخدام برنامج BCM-350 المخصص من Bistos

الخيار الأول والأكثر موثوقية هو استخدام برنامج BCM-350 من الشركة المصنعة Bistos. هذا البرنامج مصمم خصيصًا للعمل مع أجهزة مراقبة الجنين الخاصة بهم، بما في ذلك BT-350، ويفترض أنه يتعامل بشكل صحيح مع تنسيق البيانات ومقاييس الرسم.

خطوات مقترحة:

1. **الحصول على البرنامج:** حاول البحث عن برنامج BCM-350 على موقع Bistos الرسمي أو من خلال الموزعين المعتمدين. قد يتطلب الأمر التسجيل أو التواصل مع الدعم الفني للحصول على نسخة. (لاحظ أن محاولاتنا السابقة لتنزيل البرنامج مباشرة من موقع Bistos واجهت صعوبات، وقد يتطلب الأمر تعبئة نموذج أو الاتصال بهم). [2]
2. **التثبيت والتكوين:** اتبع تعليمات التثبيت الخاصة بالبرنامج. عند التكوين، تأكد من تحديد منفذ الاتصال الصحيح (RS-232C أو Ethernet) وإعدادات الاتصال (مثل معدل الباود) التي تتوافق مع جهاز BT-350. هذه الإعدادات عادة ما تكون موجودة في دليل المستخدم الخاص بالجهاز أو في إعدادات البرنامج نفسه.
3. **ضبط إعدادات الطباعة والعرض:** داخل برنامج BCM-350، ابحث عن خيارات إعدادات الطباعة والعرض. تأكد من أن مقياس الرسم (scale) وسرعة الطباعة (print speed) مضبوطة على القيم القياسية المطلوبة (مثل 1 سم/دقيقة أو 3 سم/دقيقة). البرنامج المخصص يجب أن يوفر هذه الخيارات لضمان عرض وطباعة دقيقة.

3.2. فهم بروتوكول الاتصال (التشفير) وتطوير حل مخصص (للمستخدمين المتقدمين)

إذا لم يكن برنامج BCM-350 متاحًا أو لم يحل المشكلة، فإن الخيار الأكثر تعقيدًا ولكنه الأكثر مرونة هو فهم بروتوكول الاتصال الخاص بـ BT-350 وتطوير برنامج مخصص. هذا يتطلب بعض المعرفة بالبرمجة والاتصالات التسلسلية.

لماذا هذا ضروري؟

بدون فهم كيفية

كيفية تشفير البيانات التي يرسلها الجهاز، سيكون من الصعب جدًا تصحيح مشاكل مقياس الرسم أو سرعة الطباعة بشكل دقيق.

3.2.1. استخدام برامج مراقبة المنفذ التسلسلي (Serial Port Monitors)

تسمح هذه البرامج بالتقاط وتحليل البيانات الخام التي يتم إرسالها واستقبالها عبر منفذ RS-232C. يمكن أن تكون هذه خطوة حاسمة في فهم كيفية تمثيل FHR1 و FHR2 و UC في تدفق البيانات.

أمثلة على برامج مراقبة المنفذ التسلسلي (بعضها مجاني أو يقدم نسخة تجريبية):

- **RealTerm:** أداة مجانية وقوية لمراقبة وتصحيح أخطاء المنافذ التسلسلية. [3]
- **PuTTY:** يمكن استخدامه لفتح اتصال تسلسلي وعرض البيانات الخام، على الرغم من أنه ليس مصممًا خصيصًا للمراقبة. [4]
- **Serial Port Monitor (من Eltima Software أو AGG Software):** هذه البرامج تقدم ميزات متقدمة لمراقبة وتحليل حركة المرور التسلسلية، ولكنها غالبًا ما تكون مدفوعة. [5], [6]

خطوات استخدام برامج المراقبة لفهم التشفير:

1. **التوصيل:** قم بتوصيل جهاز BT-350 بالكمبيوتر باستخدام كابل RS-232C. تأكد من استخدام كابل متصالب (null modem cable) إذا كان الجهاز يتطلب اتصالاً مباشرًا بجهاز كمبيوتر، حيث أن الكابلات القياسية قد لا تعمل. [1]
2. **تحديد المنفذ:** في نظام التشغيل Windows، يمكنك العثور على رقم منفذ COM المعين لمحول RS-232C (إذا كنت تستخدم محول USB إلى RS-232C) في إدارة الأجهزة (Device Manager) ضمن قسم 'Ports (COM & LPT)'. [7]
3. **ضبط إعدادات الاتصال الأولية:** افتح برنامج مراقبة المنفذ التسلسلي وحدد منفذ COM الصحيح. ابدأ بإعدادات الاتصال التسلسلي الشائعة:
    - **معدل الباود (Baud Rate):** 9600 (هذا هو المعدل الأكثر شيوعًا للأجهزة الطبية القديمة، ولكن قد تحتاج إلى تجربة معدلات أخرى مثل 19200 أو 38400).
    - **بتات البيانات (Data Bits):** 8
    - **التكافؤ (Parity):** None (لا يوجد)
    - **بتات التوقف (Stop Bits):** 1
    - **التحكم في التدفق (Flow Control):** None (لا يوجد)

هذه الإعدادات هي الأكثر شيوعًا (8N1) [8]. 4. **بدء المراقبة:** ابدأ التقاط البيانات في برنامج المراقبة. 5. **تشغيل الجهاز:** قم بتشغيل جهاز BT-350 وابدأ عملية مراقبة الجنين. حاول ملاحظة التغيرات في البيانات المعروضة على شاشة الجهاز ومقارنتها بالبيانات الخام التي تظهر في برنامج المراقبة. 6. **تحليل البيانات الخام:** هذه هي الخطوة الأكثر تحديًا وتتطلب الصبر والملاحظة. ابحث عن الأنماط المتكررة في تدفق البيانات: * **علامات بداية ونهاية الرسالة:** غالبًا ما تبدأ الرسائل ببايتات محددة (مثل `0x02` لـ STX) وتنتهي ببايتات أخرى (مثل `0x03` لـ ETX أو `0x0D 0x0A` لـ CR LF). هذه البايتات تساعد في تحديد

حدود كل رسالة. [9] * **طول الرسالة:** هل طول الرسائل ثابت أم متغير؟ إذا كان ثابتًا، فهذا يسهل عملية التحليل. * **تحديد قيم FHR و UC:** هذه هي النقطة الأهم. حاول ربط القيم الرقمية التي تراها على شاشة جهاز BT-350 بالبايتات المقابلة في البيانات الخام. قد تكون القيم ممثلة كأرقام ASCII (على سبيل المثال، الرقم 120 قد يظهر كـ `0x31 0x32 0x30`) أو كقيم ثنائية (على سبيل المثال، 120 قد يظهر كـ `0x78`). إذا كانت القيم ثنائية، فقد تحتاج إلى معرفة ترتيب البايتات (endianness). * **المعلومات الزمنية:** ابحث عن أي بايتات أو تسلسلات بايتات تتغير بانتظام مع مرور الوقت. هذا سيساعدك على فهم التردد الذي يتم به إرسال البيانات وكيفية ربطها بالمحور الزمني للتخطيط. * **التحقق من المجموع الاختباري (Checksum):** بعض البروتوكولات تتضمن بايتًا أو أكثر في نهاية الرسالة (قبل بايت النهاية) يستخدم للتحقق من سلامة البيانات. إذا تمكنت من تحديد خوارزمية المجموع الاختباري، فهذا يؤكد فهمك لهيكل الرسالة. [10]

3.2.2. مثال افتراضي لتفسير البيانات (للتوضيح فقط)

لنفترض، على سبيل المثال، أنك لاحظت النمط التالي في البيانات الخام (بالنظام السداسي عشري) وأنك قمت بربطه بالقيم المعروضة على الجهاز:

```
0x02 0x78 0x64 0x2A 0x05 0x03
```

قد تفسر هذه الرسالة افتراضيًا على النحو التالي:

- `0x02` : بداية الرسالة (STX)
- `0x78` : قيمة FHR1 (إذا كانت هذه قيمة FHR1 هي 120 بالرقم العشري، لأن `0x78` = 120)
- `0x64` : قيمة FHR2 (إذا كانت هذه قيمة FHR2 هي 100 بالرقم العشري، لأن `0x64` = 100)
- `0x2A` : قيمة UC (إذا كانت هذه قيمة UC هي 42 بالرقم العشري، لأن `0x2A` = 42)
- `0x05` : قد يكون هذا بايتًا للوقت أو لعلامة حركة الجنين، أو جزءًا من مجموع اختباري.
- `0x03` : نهاية الرسالة (ETX)

هذا المثال بسيط جدًا، والبروتوكولات الحقيقية غالبًا ما تكون أكثر تعقيدًا، ولكن المبدأ هو نفسه: تحديد الأنماط وتفسير البايتات.

3.2.3. تطوير برنامج مخصص (باستخدام Python كمثال)

بمجرد فهم بروتوكول الاتصال، يمكن للمستخدم كتابة برنامج خاص به لقراءة البيانات وتفسيرها وعرضها. لغة Python هي خيار ممتاز لهذا الغرض نظرًا لبساطتها وتوفر مكتبات قوية للتعامل مع المنافذ التسلسلية والرسم البياني.

الخطوات الأساسية:

1. **تثبيت مكتبة `pyserial`:** هذه المكتبة تسمح لـ Python بالتعامل مع المنافذ التسلسلية. يمكنك تثبيتها باستخدام `pip`: `bash pip install pyserial` [11]
2. **قراءة البيانات من المنفذ التسلسلي:** ```python import serial import time

إعدادات المنفذ التسلسلي (يجب أن تتطابق مع ما اكتشفته)

---

ser = serial.Serial('port='COM1', # استبدل بـ منفذ COM الصحيح الخاص بك
baudrate=9600, parity=serial.PARITY_NONE, stopbits=serial.STOPBITS_ONE,
bytesize=serial.EIGHTBITS, timeout=1 # مهلة القراءة بالثواني)

print(f

Connected to: {ser.portstr}

```
try:
    while True:
        # قراءة بايت واحد في كل مرة أو حتى علامة نهاية الرسالة
        # هذا الجزء يعتمد بشكل كبير على بروتوكول الجهاز الخاص بك
        # كمثال، لنفترض أننا نقرأ سطرًا ينتهي بـ \n
        line = ser.readline().decode("ascii").strip()
        if line:
            print(f"Received: {line}")
            # هنا يأتي جزء تحليل البيانات (parsing)
            # لنفترض أن السطر يحتوي على FHR1, FHR2, UC مفصولة بفاصلة
            try:
                # مثال بسيط جداً: "FHR1:140,FHR2:130,UC:50"
                parts = line.split(",")
                fhr1 = int(parts[0].split(":")[1])
                fhr2 = int(parts[1].split(":")[1])
                uc = int(parts[2].split(":")[1])

                print(f"Parsed - FHR1: {fhr1}, FHR2: {fhr2}, UC: {uc}")

                # هنا يمكنك تخزين البيانات في قائمة أو قاعدة بيانات
                # ثم استخدامها للرسم البياني

            except (IndexError, ValueError) as e:
                print(f"Error parsing line: {line} - {e}")

        time.sleep(0.1) # انتظر قليلاً قبل القراءة التالية

except KeyboardInterrupt:
    print("Exiting program.")
finally:
    ser.close()
    print("Serial port closed.")
```
ملاحظة هامة: الجزء الخاص بـ `line = ser.readline().decode("ascii").strip()` هو مجرد مثال توضيحي. يجب تعديله ليتناسب تمامًا مع تنسيق البيانات الفعلي الذي يرسله جهاز BT-350 الخاص بك بعد تحليله باستخدام برامج مراقبة المنفذ التسلسلي.

1. **تحليل البيانات (Parsing):** بعد قراءة البيانات الخام، الخطوة التالية هي تحليلها (parsing) لاستخراج قيم FHR1 و FHR2 و UC. هذا يتضمن تحويل البايتات الخام إلى قيم رقمية قابلة للاستخدام. يعتمد هذا بشكل كامل على بروتوكول الاتصال الذي اكتشفته. قد تحتاج إلى:
    - تحويل الأرقام من النظام السداسي عشري أو الثنائي إلى العشري.
    - تجميع بايتات متعددة لتكوين قيمة واحدة (على سبيل المثال، بايتين لـ FHR1).
    - التعامل مع أي علامات خاصة أو بادئات/نهايات للرسائل.

2. **عرض البيانات ورسمها بيانيًا (Visualization):** بمجرد استخراج القيم، يمكنك عرضها على الشاشة أو رسمها بيانيًا. مكتبة `matplotlib` في Python ممتازة لهذا الغرض. يمكنك تحديث الرسم البياني في الوقت الفعلي أو حفظ البيانات ورسمها لاحقًا.

```python import matplotlib.pyplot as plt import matplotlib.animation as animation from collections import deque
```

تأكد من أنك قمت بتثبيت pyserial و matplotlib

---

pip install pyserial matplotlib

... (كود إعداد المنفذ التسلسلي ser كما في المثال السابق)

إعداد البيانات للرسم البياني

---

fhr1_data = # عدد النقاط التي ستظهر على الرسم البياني في نفس الوقت max_points = 500 deque([0.0]*max_points, maxlen=max_points) fhr2_data = deque([0.0]*max_points, maxlen=max_points) uc_data = deque([0.0]*max_points, maxlen=max_points) time_data = deque([0.0]*max_points, maxlen=max_points)

fig, (ax1, ax2) = plt.subplots(2, 1, sharex=True) line1, = ax1.plot(time_data, fhr1_data, label=\'FHR1\', color=\'blue\') line2, = ax1.plot(time_data, fhr2_data, label=\'FHR2\', color=\'green\') line3, = ax2.plot(time_data, uc_data, label=\'UC\', color=\'red\')

ax1.set_ylabel(\'FHR (bpm)\') ax2.set_ylabel(\'UC (mmHg)\') ax2.set_xlabel(\'Time (s)\') ax1.set_title(\'Fetal Monitor Data\') ax1.legend() ax2.legend() ax1.set_ylim(50, 200) # نطاق FHR النموذجي ax2.set_ylim(0, 100) # نطاق UC النموذجي plt.grid(True)

start_time = time.time()

def update(frame): global start_time if ser.in_waiting > 0: try: # هذا الجزء يجب أن يتطابق مع تحليل البيانات الخاص بك # لنفترض أننا نستقبل قيم FHR1, FHR2, UC مباشرة # مثال: قراءة 6 بايتات (2 لـ FHR1, 2 لـ FHR2, 2 لـ UC) # هذا مجرد مثال، قد تحتاج إلى قراءة أطول أو أقصر بناءً على بروتوكولك raw_data = ser.read(6) if len(raw_data) == 6: # مثال افتراضي: تحويل بايتين إلى قيمة عدد صحيح # يجب تعديل هذا بناءً على كيفية تشفير جهازك للقيم fhr1_val = int.from_bytes(raw_data[0:2], byteorder=\'big\') fhr2_val = int.from_bytes(raw_data[2:4], byteorder=\'big\') uc_val = int.from_bytes(raw_data[4:6], byteorder=\'big\')

```
            fhr1_data.append(fhr1_val)
            fhr2_data.append(fhr2_val)
            uc_data.append(uc_val)
            time_data.append(time.time() - start_time)

            line1.set_data(time_data, fhr1_data)
            line2.set_data(time_data, fhr2_data)
            line3.set_data(time_data, uc_data)

            ax1.set_xlim(time_data[0], time_data[-1])
            ax2.set_xlim(time_data[0], time_data[-1])

    except Exception as e:
        print(f"Error reading or parsing serial data: {e}")
return line1, line2, line3
```

``` ani = animation.FuncAnimation(fig, update, interval=100, blit=True) plt.show()

**معالجة مشكلة التخطيط الطويل وسرعة الطباعة في برنامج مخصص:**

- **مقياس الرسم (Scaling):** في برنامجك المخصص، لديك تحكم كامل في مقياس الرسم. يمكنك ضبط المحور الزمني (X-axis) ليتوافق مع المقياس القياسي (مثل 1 سم/دقيقة). إذا كان الجهاز يرسل نقطة بيانات كل ثانية، وتريد أن تمثل كل دقيقة 1 سم على الرسم البياني، فيمكنك حساب عدد النقاط التي يجب أن تظهر في 1 سم وتعديل المحور X وفقًا لذلك.
- **سرعة الطباعة:** عند حفظ الرسم البياني كصورة (PNG/JPEG) أو ملف PDF، يمكنك تحديد الأبعاد (بالبكسل أو البوصة) لضمان أن المقياس صحيح. على سبيل المثال، إذا كنت تريد أن
```

يكون 1 سم على الورق، يمكنك تحديد أن 1 سم في الرسم البياني يتوافق مع عدد معين من البكسلات عند الحفظ.

3.3. البحث عن برامج طرف ثالث (Third-Party Software)

إذا كان تطوير برنامج مخصص صعبًا، يمكن البحث عن برامج طرف ثالث مصممة للتعامل مع بيانات أجهزة مراقبة الجنين عبر RS-232C. قد لا تكون هذه البرامج مجانية، ولكنها قد توفر حلاً جاهزًا. عند البحث، ابحث عن برامج تدعم الاتصال التسلسلي وتوفر خيارات لضبط مقياس الرسم وإعدادات الطباعة.

4. الخلاصة والتوصيات

إن ربط جهاز Bistos BT-350 بالكمبيوتر وعرض التخطيط بشكل صحيح يتطلب فهمًا لكيفية عمل الجهاز وبروتوكول الاتصال الخاص به. المشكلة التي تواجهها تتعلق على الأرجح بمقياس الرسم وتفسير البيانات الزمنية في البرنامج الذي تستخدمه.

التوصيات الرئيسية:

1. **الأولوية لبرنامج BCM-350:** حاول بكل الطرق الممكنة الحصول على برنامج BCM-350 من Bistos. هذا هو الحل الأكثر احتمالاً للعمل بشكل صحيح دون الحاجة إلى فهم عميق للبروتوكول.
2. **استخدام برامج مراقبة المنفذ التسلسلي:** إذا لم تتمكن من الحصول على BCM-350، فاستخدم برامج مراقبة المنفذ التسلسلي (مثل RealTerm) لالتقاط البيانات الخام من جهازك. هذه الخطوة حاسمة لفهم

بروتوكول الاتصال (التشفير) الخاص بجهازك. 3. **تطوير حل مخصص (إذا لزم الأمر):** إذا تمكنت من فهم بروتوكول الاتصال، يمكنك كتابة برنامج بسيط (باستخدام Python مثلاً) لقراءة البيانات وتحليلها وعرضها بيانيًا بالمقياس الصحيح. هذا يمنحك التحكم الكامل في كيفية عرض البيانات وطباعتها. 4. **معايرة الطباعة:** تأكد من أن أي برنامج تستخدمه (سواء كان BCM-350 أو برنامجًا مخصصًا) يسمح بمعايرة مقياس الطباعة لضمان أن 1 سم على الشاشة أو في الإخراج يتوافق مع 1 سم على الورق.

نأمل أن يساعدك هذا الدليل في حل مشكلتك. تذكر أن فهم بروتوكول الاتصال هو المفتاح للتحكم الكامل في البيانات المعروضة.

5. المراجع

Dolphitonic. (2013, November 13). *Manuel LOGIVIEL BCM 350 ECOTWIN*. [PDF]. [1] Available at: ww.dolphitonic.com/img/cms/Manuel%20LOGIVIEL%20BCM%20350%20ECOTWIN.pdf

[2] Bistos. (n.d.). *Download*. Available at: https://www.bistos.co.kr/download/ [3] RealTerm. (n.d.). *RealTerm: Serial/TCP Terminal*. Available at: https://realterm.sourceforge.io/ [4] PuTTY. (n.d.). *PuTTY: A free SSH and Telnet client*. Available at: https://www.putty.org/ [5] Eltima Software. (n.d.). *Serial Port Monitor*. Available at: https://www.eltima.com/products/serial-port-monitor/ [6] AGG Software. (n.d.). *Serial Port Monitor*. Available at: https://www.aggsoft.com/serial-port-monitor.htm [7] Microsoft. (n.d.). *Open Device Manager*. Available at: https://support.microsoft.com/en-us/windows/open-device-manager-a7257945-427c-b3e2-167d-248279522143 [8] Contec. (n.d.). *Serial communication Basic Knowledge - RS-232C/RS-422A/485*. Available at: https://www.contec.com/support/basic-knowledge/daq-control/serial-communicatin/ [9] Circuit Digest. (2020, May 1). *RS232 Serial Communication Protocol*. Available at: https://circuitdigest.com/article/rs232-serial-communication-protocol-basics-specifications [10] ResearchGate. (2018, March 29). *(PDF) Development of a Smart Mobile Data Module for Fetal Monitoring in E-Healthcare*. Available at: 23967933_Development_of_a_Smart_Mobile_Data_Module_for_Fetal_Monitoring_in_E-Healthcare [11] PySerial. (n.d.). *PySerial: Python Serial Port Access*. Available at: https://pyserial.readthedocs.io/en/latest/ [12] Matplotlib. (n.d.). *Matplotlib: /Visualization with Python*. Available at: https://matplotlib.org